ELC 2137 Lab 08: 4-digit Display

Maddie Vorhies

October 20, 2020

Summary

My goal in this lab was to create a 4-digit display and add the ability to switch between hexadecimal and decimal (BCD) output on my Basys3 board. To do this, I used the sseg decoder, 11-bit circuit, and the mux2 from the previous lab. I modified the mux2 to include parameters in order to create flexible and reusable modules. After my mux2 was built, I created the mux4 and the anode decoder. The anode decoder allows you to switch between the four output digits. Once these were all built a created the sseg4 module to put all of these different part together. Lastly I created a manual that made the code much more organized and easier to read. I was then able to program my board and test the given inputs given in the lab. All of my test results were successful. My board successfully produced the correct outputs.

Results

Time (ns)	in0	in1	sel	output
0	0110	1001	0	0110
10	0110	1001	1	1001
20	0011	1011	0	0011
30	0011	1011	1	1011

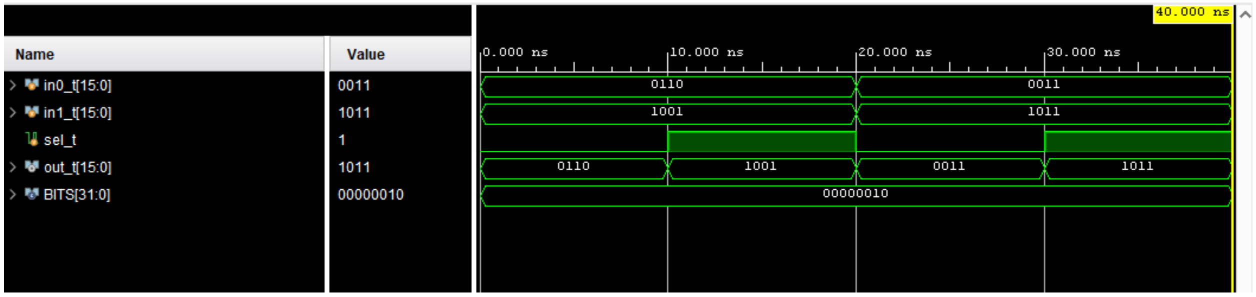

Figure 1: Simulation Waveform and ERT of mux2

Code

```
‘timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////
```

```
// Company:
// Engineer:
//
// Create Date: 10/15/2020 11:37:17 AM
// Design Name:
// Module Name: mux2
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   ////////////////////////////////////////////////////////////////////////////////


module mux2
    #(parameter BITS = 4)
    (
    input [BITS-1:0] in0,
    input [BITS-1:0] in1,
    input sel,
    output [BITS-1:0] out
    );

    assign out = sel ? in1 : in0;


endmodule
```

```
`timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/15/2020 12:09:07 PM
// Design Name:
// Module Name: mux4
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
```

```
// Revision 0.01 - File Created
// Additional Comments:
//
//
   ////////////////////////////////////////////////////////////////////////////////



module mux4
   #(parameter BITS = 4)
   (
   input [BITS-1:0] in0,
   input [BITS-1:0] in1,
   input [BITS-1:0] in2,
   input [BITS-1:0] in3,
   input [1:0] sel,
   output reg [BITS-1:0] out
   );

   always @*
      case (sel)

         2'b00: out = in0;
         2'b01: out = in1;
         2'b10: out = in2;
         default: out = in3;

      endcase

endmodule
```

```
`timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/15/2020 12:38:36 PM
// Design Name:
// Module Name: an_decoder
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   ////////////////////////////////////////////////////////////////////////////////
```

```
module an_decoder(
    input [1:0] in,
    output reg [3:0] out
    );

    always @*
       case (in)

          2'b00: out = 4'b1110;
          2'b01: out = 4'b1101;
          2'b10: out = 4'b1011;
          2'b11: out = 4'b0111;
       endcase

endmodule
```

```
`timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/15/2020 01:05:29 PM
// Design Name:
// Module Name: sseg4
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   ////////////////////////////////////////////////////////////////////////////////



module sseg4(
    input [15:0] data,
    input hex_dec,
    input sign,
    input [1:0] digit_sel,
    output [6:0] seg,
    input dp,
    input [3:0] an
    );
    wire [15:0] bcd11_out;
    wire [15:0] out_mux2;
    wire [3:0] mux4out;
    wire [6:0] dec_out;
    wire [3:0] andec_out;
    wire mux2_sel;

    bcd11 bcd (
       .B(data[10:0]),
       .ones(bcd11_out[3:0]),
       .tens(bcd11_out[7:4]),
       .hundreds(bcd11_out[11:8]),
       .thousands(bcd11_out[15:12])
    );

    mux2 #(.BITS(16)) mux2_0 (
       .in1(data[15:0]),
       .in0(bcd11_out[15:0]),
       .out(out_mux2[15:0]),
       .sel(hex_dec)
   );

   mux4 #(.BITS(16)) mux4_0 (
      .sel(digit_sel[1:0]),
      .out(mux4out[3:0]),
      .in3(out_mux2[15:12]),
      .in2(out_mux2[11:8]),
      .in1(out_mux2[7:4]),
      .in0(out_mux2[3:0])
   );

   sseg_decoder sseg_dec (
      .num(mux4out[3:0]),
      .sseg(dec_out[6:0])
   );

   assign mux2_sel = sign & ~andec_out[3];
   assign an = andec_out;
   assign dp = 1'b1;

   mux2 #(.BITS(16)) mux2_1 (
      .in1(7'b0111111),
      .in0(dec_out[6:0]),
      .sel(mux2_sel),
      .out(seg[6:0])
   );

   an_decoder an_dec (
      .in(digit_sel[1:0]),
      .out(andec_out[3:0])
   );

endmodule
```

```
‘timescale 1ns / 1ps
//
   ////////////////////////////////////////////////////////////////////////////////////

// Company:
// Engineer:
//
// Create Date: 10/20/2020 08:21:39 PM
// Design Name:
// Module Name: sseg4_manual
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//
   ////////////////////////////////////////////////////////////////////////////////////



module sseg4_manual(
    input [15:0] sw,
    output [6:0] seg,
    output dp,
    output [3:0] an
    );

    sseg4 sseg4_man (
       .data({4'b0000, sw[11:0]}),
       .hex_dec(sw[15]),
       .sign(sw[14]),
       .digit_sel(sw[13:12]),
       .seg(seg[6:0]),
       .dp(dp),
       .an(an[3:0])
    );


endmodule
```